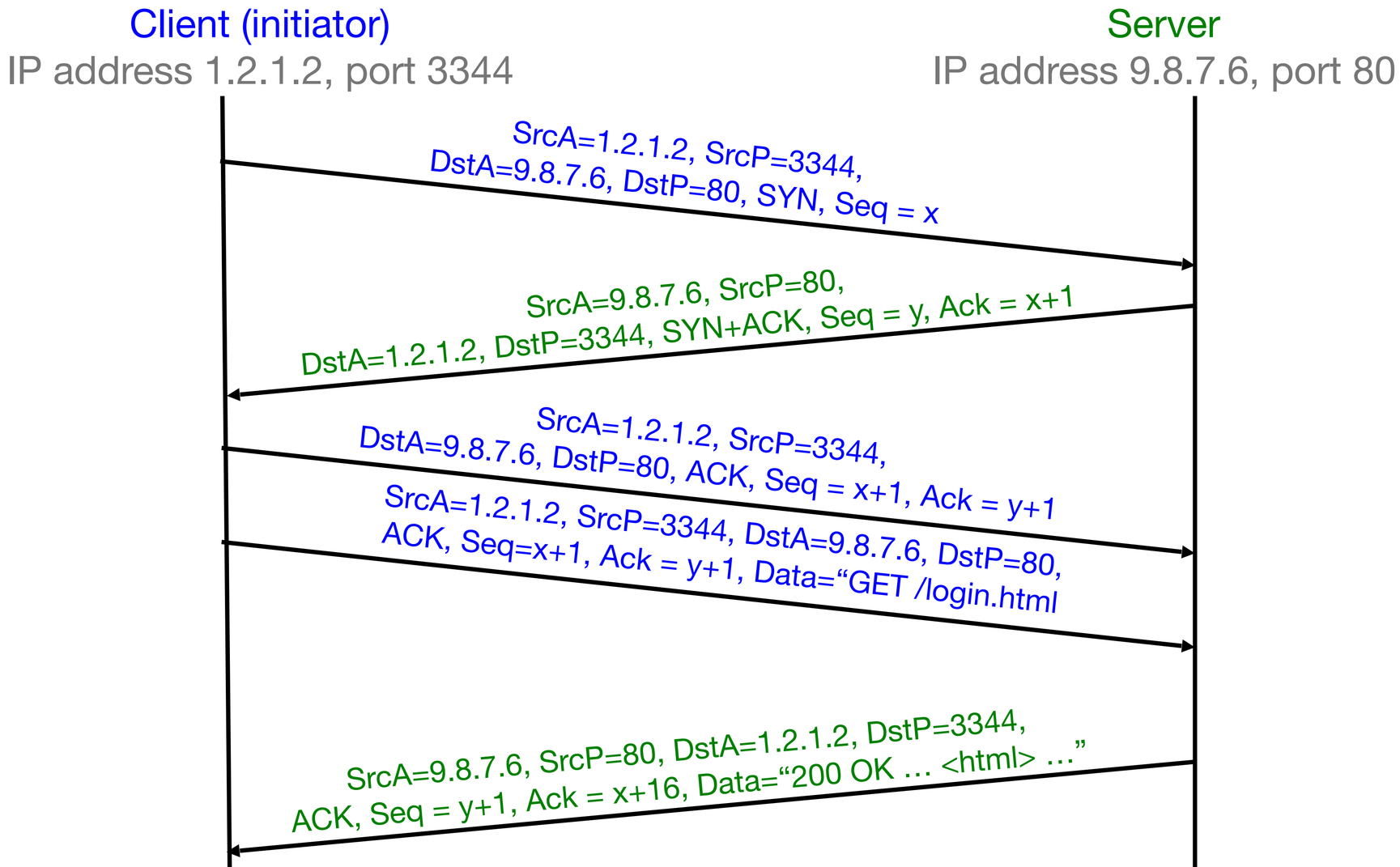
Lecture 19: Network Attacks

https://cs161.org

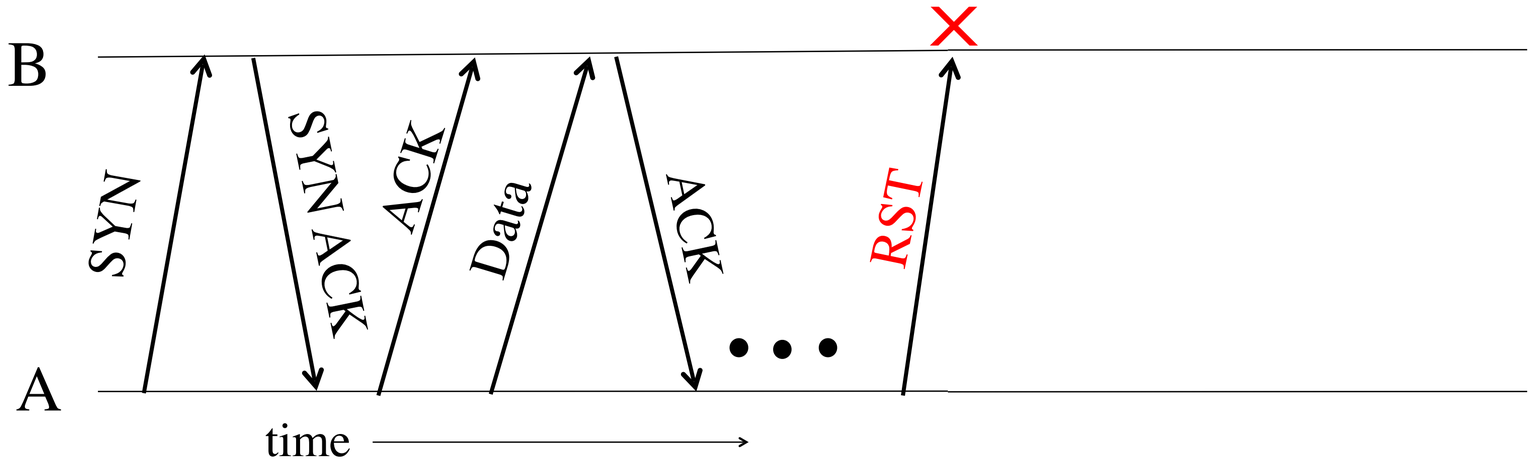
Announcements

- Project 2 design doc due today
- Networking tutorial, Saturday 3/7, 5-7pm, in HP Auditorium (306 Soda)

TCP Conn. Setup & Data Exchange



Abrupt Termination



- If A sends a TCP packet with RST flag to B and sequence number fits, connection is terminated
  - Unilateral, and takes effect immediately

TCP Threat: Disruption aka RST injection

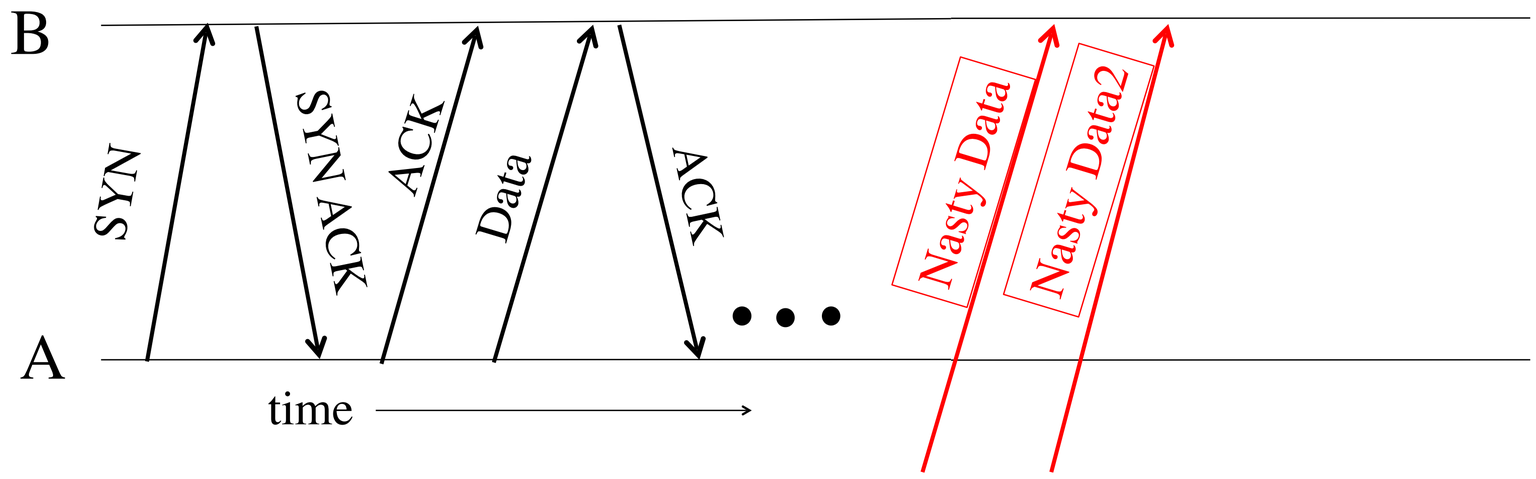
- The attacker can inject RST packets and block connection
  - TCP clients must respect RST packets and stop all communication

- Who uses this?
  - China: The Great Firewall does this to TCP requests
  - A long time ago: Comcast, to block BitTorrent uploads
  - Some intrusion detection systems: To hopefully mitigate an attack in progress

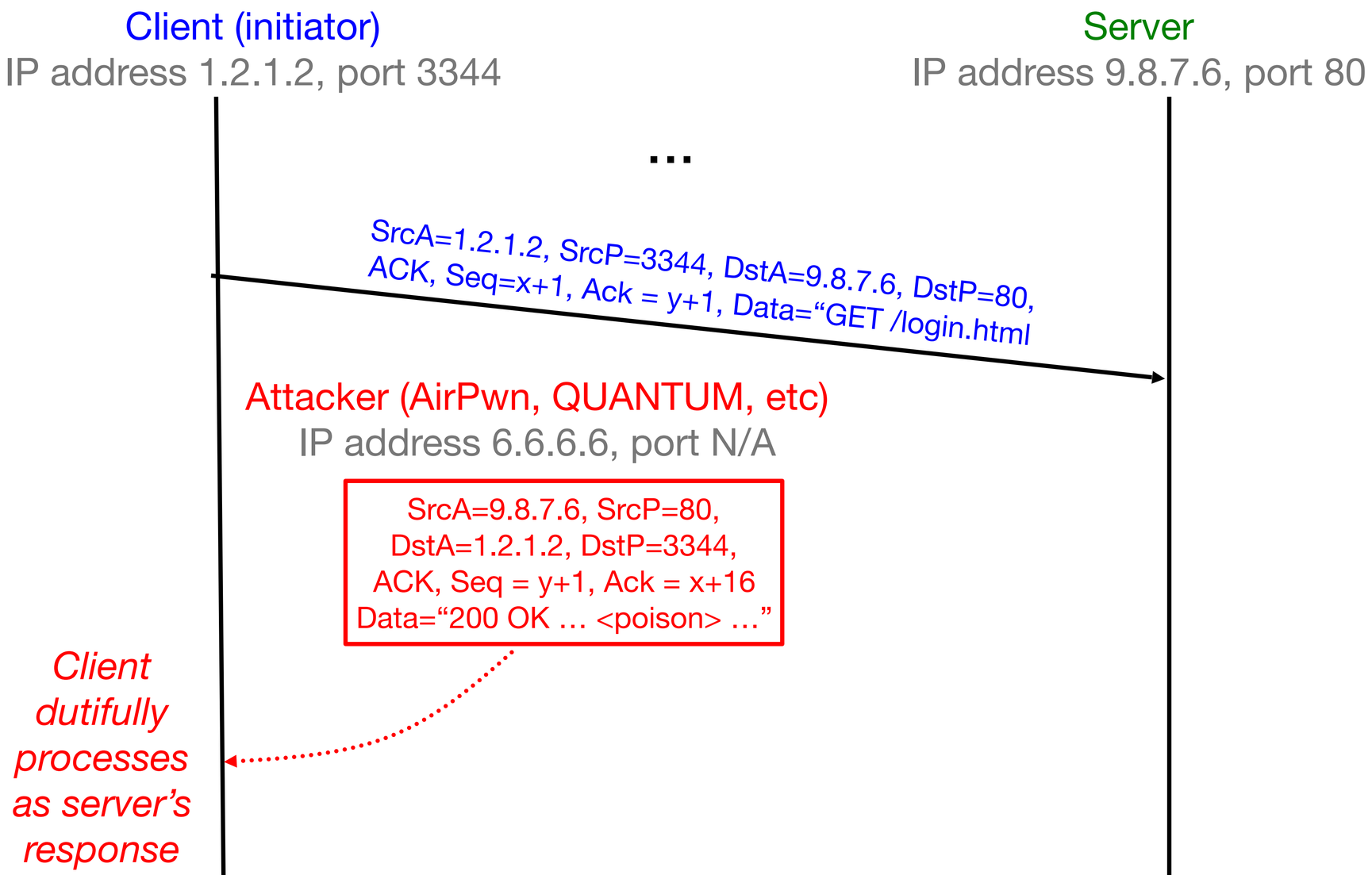
Discuss with a partner: Who can do RST injection? (a) off-path attacker, (b) on-path attacker, (c) man-in-the-middle

TCP Threat: Data Injection

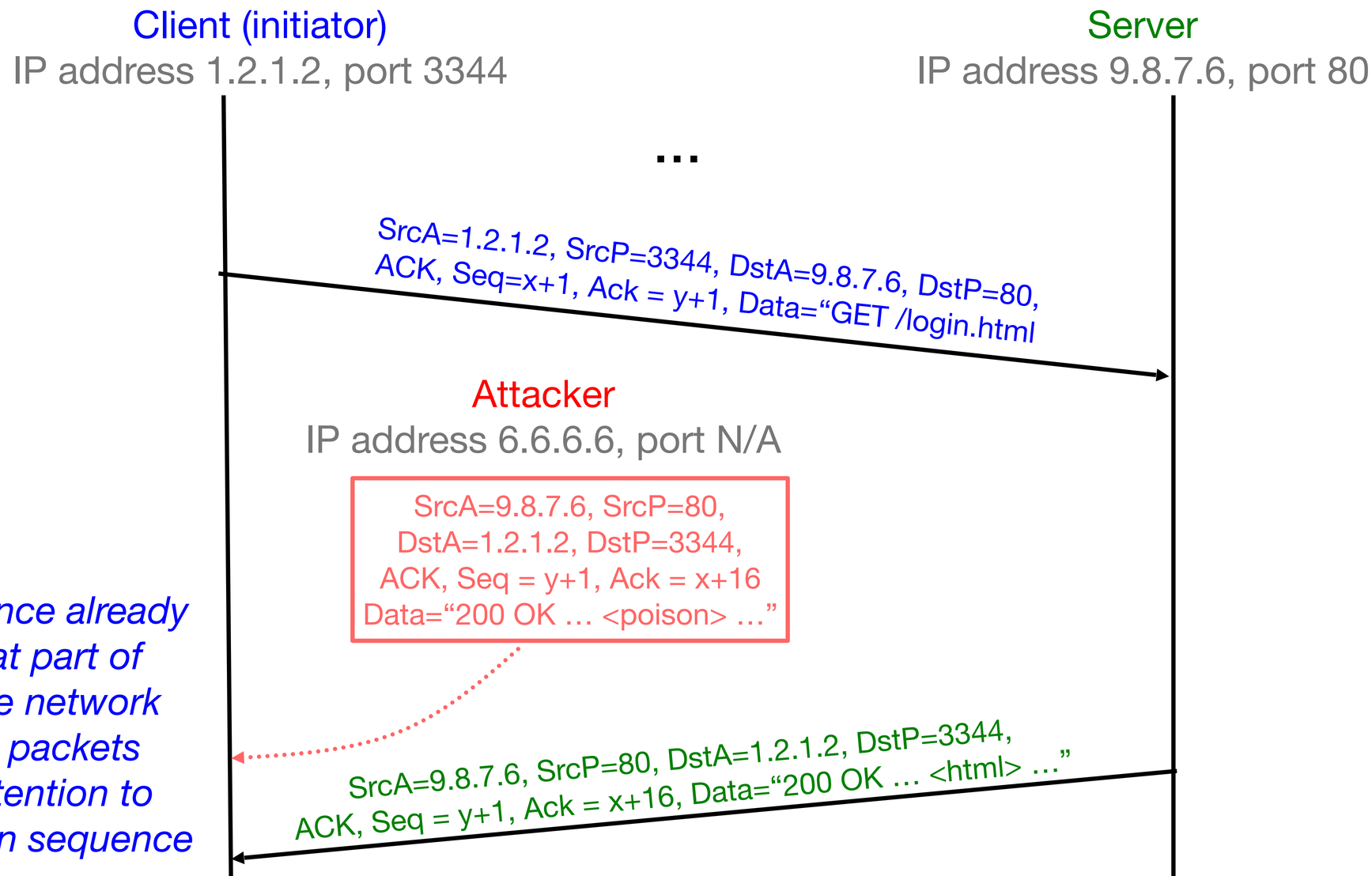
- If attacker knows ports & sequence numbers (e.g., on-path attacker), attacker can inject data into any TCP connection
  - Receiver B is *none the wiser!*
- Termed TCP connection hijacking (or "*session hijacking*")
  - A general means to take over an already-established connection!
- We are toast if an attacker can see our TCP traffic!
  - Because then they immediately know the port & sequence numbers



TCP Data Injection



TCP Data Injection



Client ignores since already processed that part of bytestream: the network can duplicate packets so only pay attention to the first version in sequence

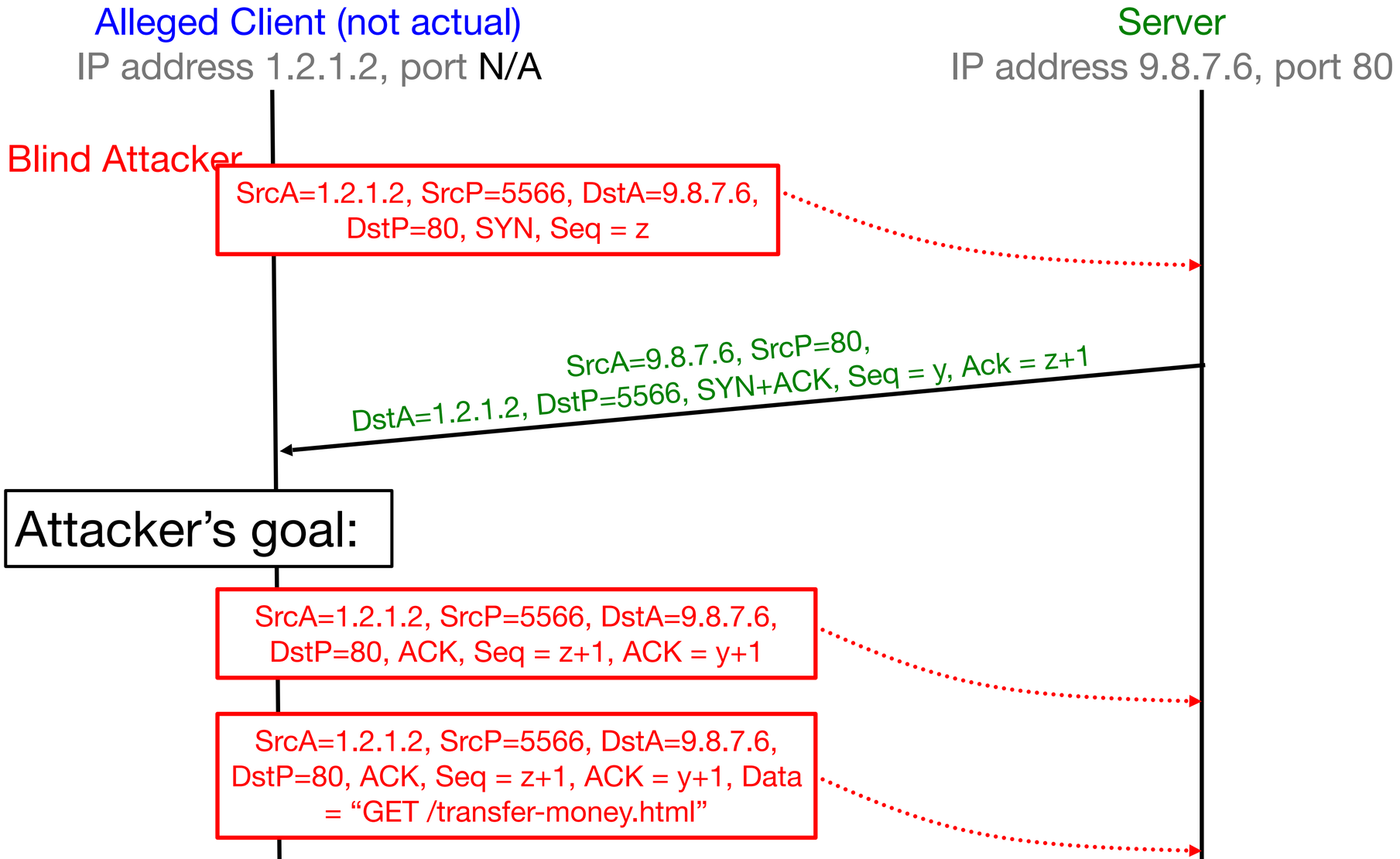
TCP Threat: Blind Hijacking

- Is it possible for an off-path attacker to inject into a TCP connection even if they can't see our traffic?
- YES: if somehow they can infer or guess the port and sequence numbers

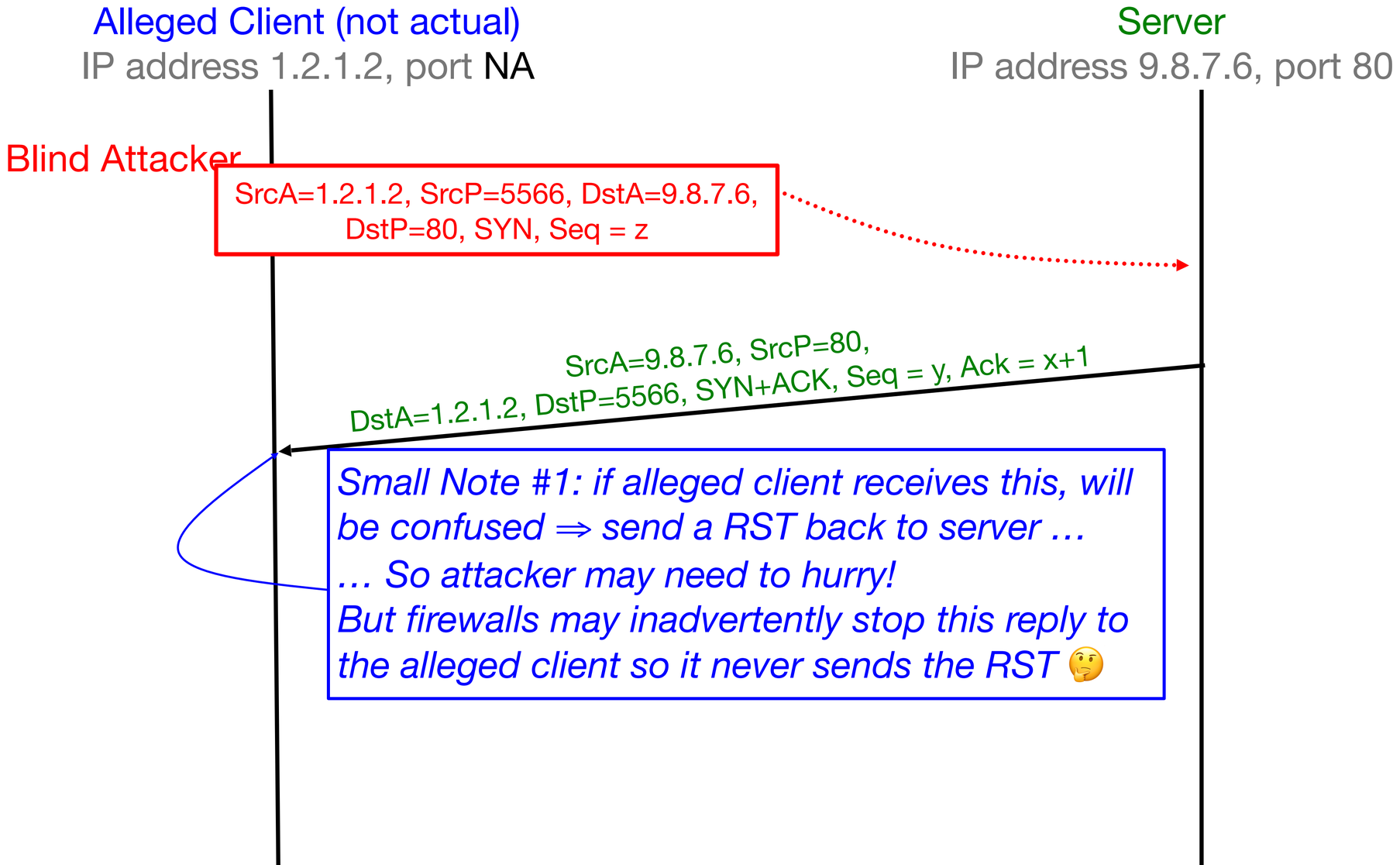
TCP Threat: Blind Spoofing

- Is it possible for an off-path attacker to create a fake TCP connection, even if they can't see responses?
- Yes if somehow they can infer or guess the TCP initial sequence numbers
- Why would an attacker want to do this?
  - Perhaps to leverage a server's trust of a given client as identified by its IP address
  - Perhaps to frame a given client so the attacker's actions during the connections can't be traced back to the attacker

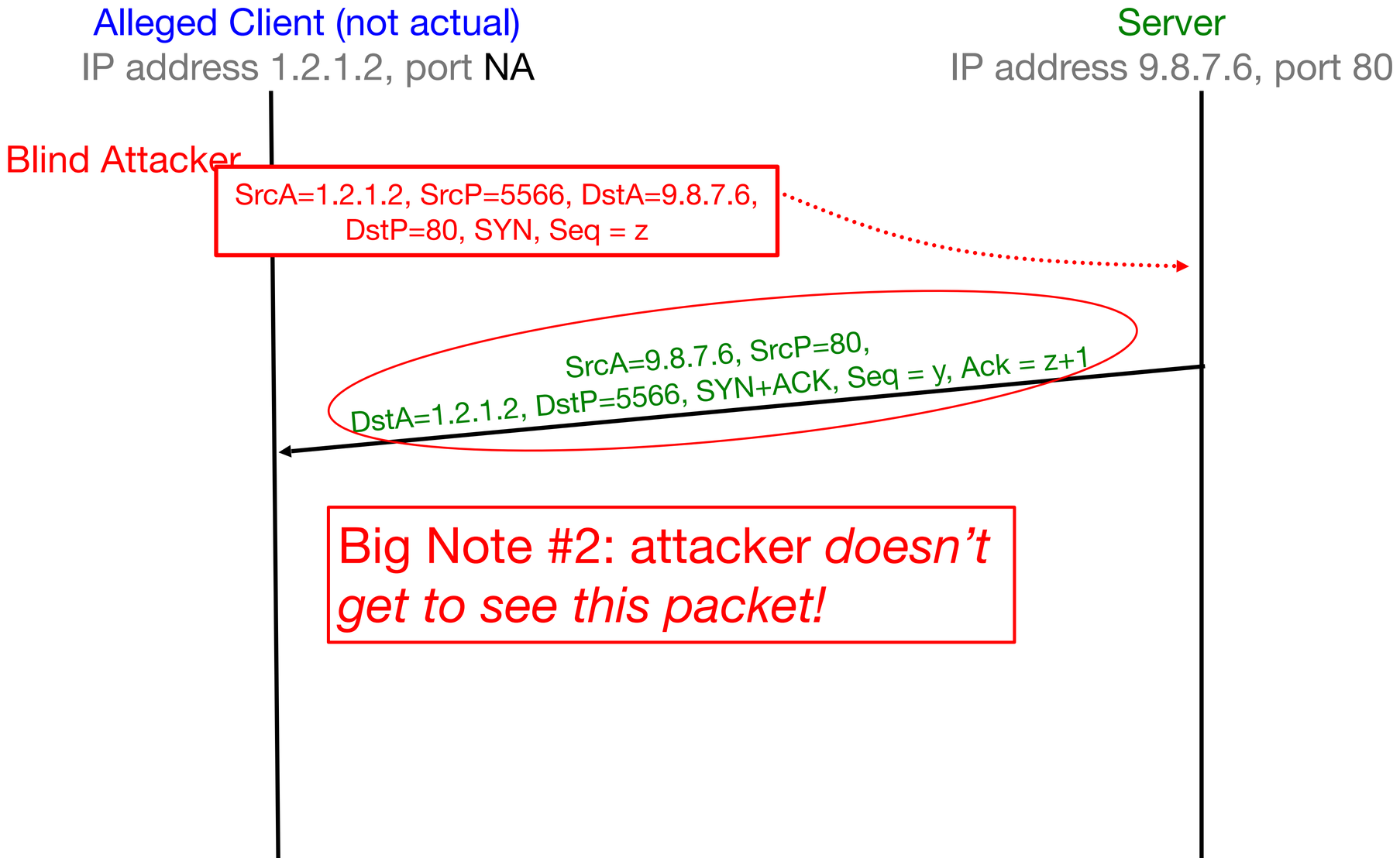
Blind Spoofing on TCP Handshake



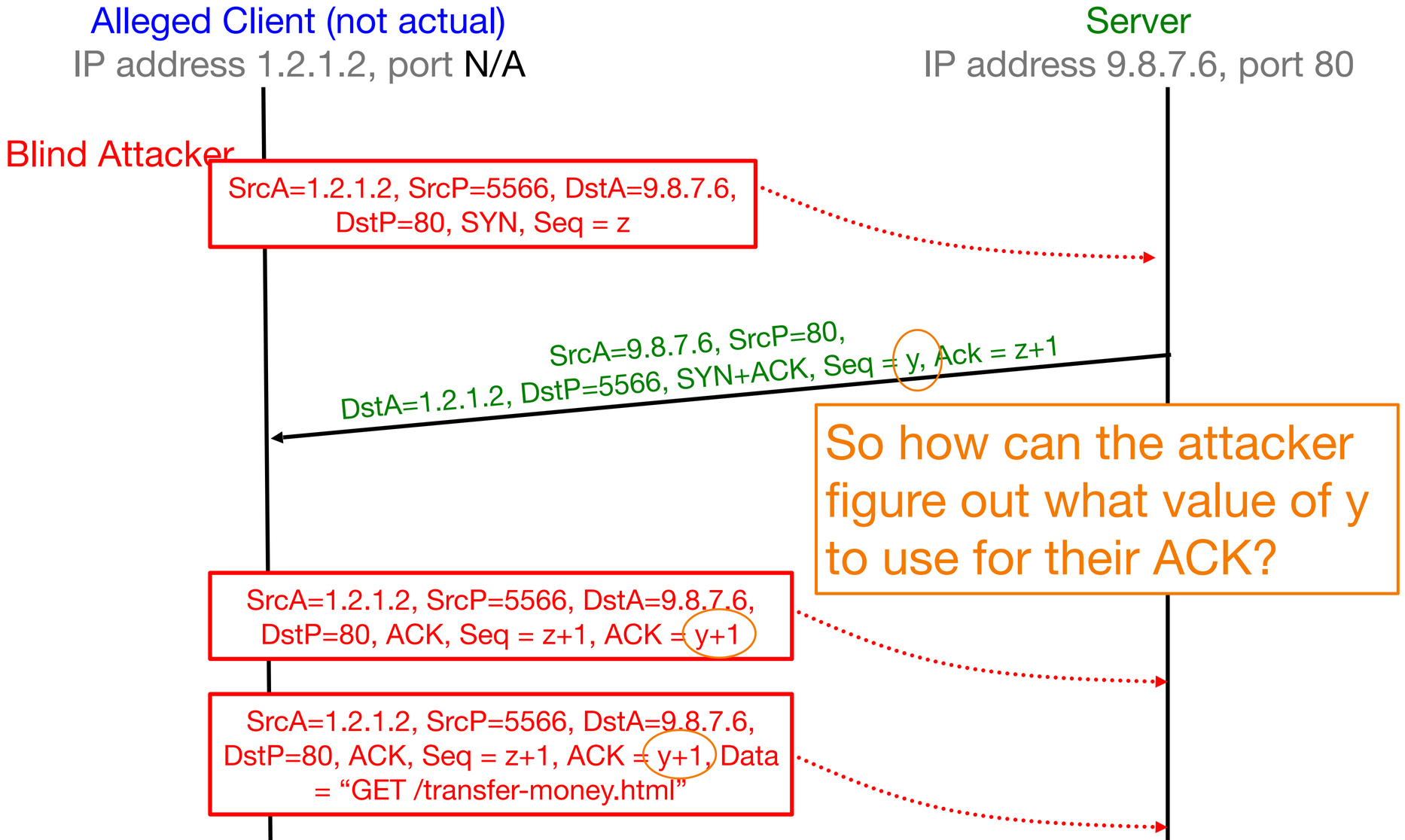
Blind Spoofing on TCP Handshake



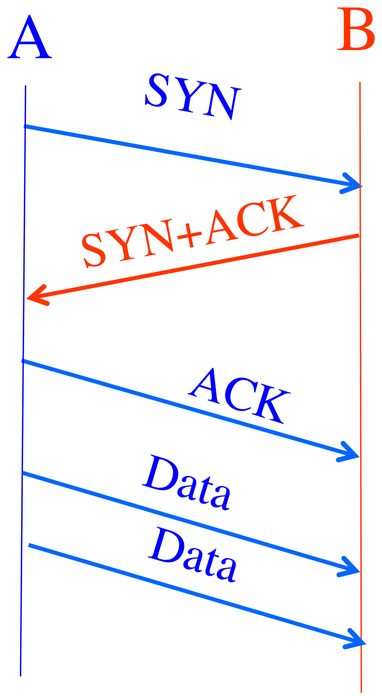
Blind Spoofing on TCP Handshake



Blind Spoofing on TCP Handshake



Reminder: Establishing a TCP Connection



Each host tells its *Initial Sequence Number* (ISN) to the other host.

(Spec says to pick based on local clock)

Hmm, any way for the attacker to know *this?*

How Do We Fix This?

Use a (Pseudo)-*Random* ISN

Sure – make a non-spoofed connection *first*, and see what server used for ISN y then!

Summary of TCP Security Issues

- An attacker who can observe your TCP connection can manipulate it:
  - Forcefully terminate by forging a RST packet
  - Inject (spoof) data into either direction by forging data packets
  - Works because they can include in their spoofed traffic the correct sequence numbers (both directions) and TCP ports
  - Remains a major threat today
- Blind spoofing no longer a threat
  - Due to randomization of TCP initial sequence numbers

Ghost of blind spoofing...

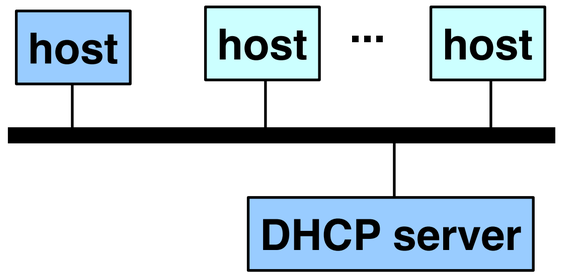
- CVE-2016-5696
  - "Off-Path TCP Exploits: Global Rate Limit Considered Dangerous" Usenix Security 2016 https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/cao
- Key idea:
  - RFC 5961 added some global rate limits that acted as an ***information leak***:
    - Could determine if two hosts were communicating on a given port
    - Could determine if your guess at the sequence number is “in window”
  - Once you get the sequence #s, you can then inject arbitrary content into the TCP stream
- Fixed today

DNS Service

- Runs Domain Name Servers
- Translates domain names google.com to IP addresses
- When user browser wants to contact google.com, it first contacts a DNS to find out the IP address for google.com and then sends a packet to that IP address
- More soon..

LAN Bootstrapping: DHCP

- New host doesn't have an IP address yet
  - So, host doesn't know what source address to use
- Host doesn't know *who to ask* for an IP address
  - So, host doesn't know what destination address to use
- Solution: shout to "**discovery**" server that can help
  - Broadcast a server-discovery message (layer 2)
  - Server(s) sends a reply offering an address

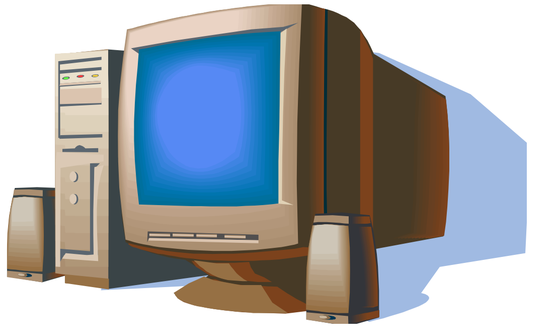


Dynamic Host Configuration Protocol

new client

DHCP discover (broadcast)

DHCP offer



DHCP server

"**offer**" message includes IP address, DNS server, "gateway router", and how long client can have these ("lease" time)

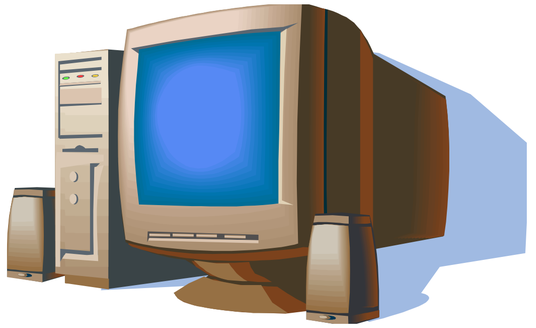
DNS server = system used by client to map hostnames like `gmail.com` to IP addresses like `74.125.224.149`

Gateway router = router that client uses as the first hop for all of its Internet traffic to remote hosts

Dynamic Host Configuration Protocol

new client

DHCP discover (broadcast)



DHCP server

DHCP offer

DHCP request (broadcast)

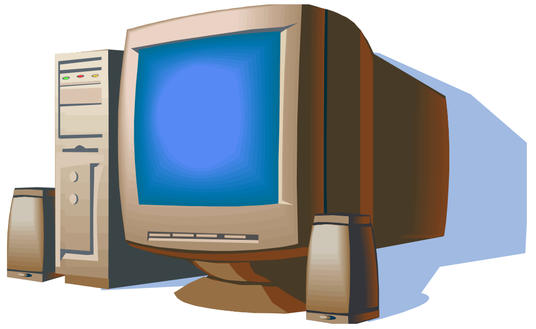
DHCP ACK

"**offer**" message includes IP address, DNS server, "gateway router", and how long client can have these ("lease" time)

Dynamic Host Configuration Protocol

new client

DHCP discover (broadcast)



DHCP server

DHCP offer

"**offer**" message includes IP address, DNS server, "gateway router", and how long client can have these ("lease" time)

DHCP request (broadcast)

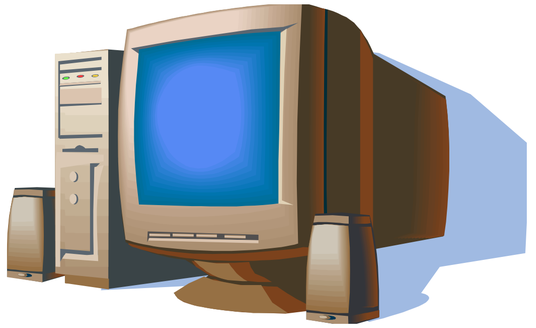
Threats?

DHCP ACK

Dynamic Host Configuration Protocol

new client

DHCP discover (broadcast)



DHCP server

DHCP offer

DHCP request (broadcast)

DHCP ACK

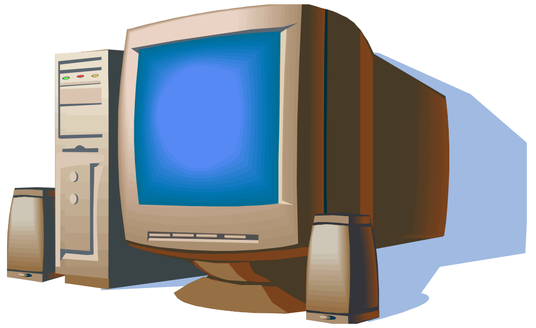
Local attacker on same subnet can **hear** new host's DHCP request

"**offer**" message includes IP address, DNS server, "gateway router", and how long client can have these ("lease" time)

Dynamic Host Configuration Protocol

new client

DHCP discover (broadcast)



DHCP server

DHCP offer

DHCP request (broadcast)

DHCP ACK

"**offer**" message includes IP address, DNS server, "gateway router", and how long client can have these ("lease" time)

Attacker can **race** the actual server; if attacker wins, replaces DNS server and/or gateway router

DHCP Threats

- Substitute a fake DNS server
  - Redirect any of a host's lookups to a machine of attacker's choice
- Substitute a fake gateway router
  - Intercept all of a host's off-subnet traffic (even if not preceded by a DNS lookup)
  - Relay contents back and forth between host and remote server and modify however attacker chooses
- An invisible Man In The Middle (MITM)
  - Victim host has no way of knowing it's happening
- How can we fix this?

Hard, because we lack a ***trust anchor***

Takeaways

- Broadcast protocols inherently at risk of local attacker spoofing
- When initializing, systems are particularly vulnerable because they can lack a trusted foundation to build upon
  - Tension between ***wiring in trust*** vs. ***flexibility and convenience***
- MITM attacks insidious because no indicators they're occurring

- DNS translates www.google.com to 74.125.25.99
- It's a performance-critical distributed database.
- DNS security is critical for the web.
  (Same-origin policy assumes DNS is secure.)
- Analogy: If you don't know the answer to a question, ask a friend for help (who may in turn refer you to a friend of theirs, and so on).
- Based on a notion of hierarchical trust: we trust . for everything, com. for any com, google.com. for everything google…

DNS Lookups via a *Resolver*

Host at **`xyz.poly.edu`** wants IP address for **`eecs.mit.edu`**